Delegatable Anonymous Credentials for Voting

Tommy Heng, Ellen Wang, Ioannis Kaklamanis

March 2022

1 Introduction

The most prevalent forms of democracy in the world, across governmental systems and organizations, belong within one of two categories: direct democracy and representational democracy. Direct democracy is the most direct form of voting. Individuals vote on each issue directly. However, this system does not scale with the size of the community. With larger voter bases, this system becomes infeasible. Representational democracies are the most widely used form of voting. Individuals vote for representatives who serve as experts in policy. These representatives then vote on issues on behalf of the voters. However, this system is prone to corruption and often fails to serve the voters' best interests [Sch15].

Liquid democracy proposes to solve the issues with representational and direct democracy with voter choice. In liquid democracy, voters are given full control over vote delegation. Individuals chose whether to vote directly or rely on a representative. In this fashion, liquid democracy increases voter turnout by enabling voters who were previously discouraged by their lack of choice to embody a more active role in policy making [Sch15].

We propose a system that would allow for the flexibility for liquid democracy while preserving anonymity, both to the representative receiving your vote and to the authority that is counting your votes.

1.1 The Issue of Anonymity

Traditional credential methods follow a certificate and signature model, where parties are identified using a constant alias. In the traditional credential model, an unidentified party generates a public and private key pair and applies for a digital certificate - an electronic document that verifies their identity. Once certified, the verified individual signs their certificate with their private key and sends this signed certificate with their messages, allowing recipients to verify their identity by decrypting the certificate using the sender's public key.

This method of authentication continuously exposes the identity of the sender by using the same public key for a given individual. Adversaries can track the traffic for a specific public key and observe an individual's behaviors. This credential system is especially problematic for sensitive information, such as with

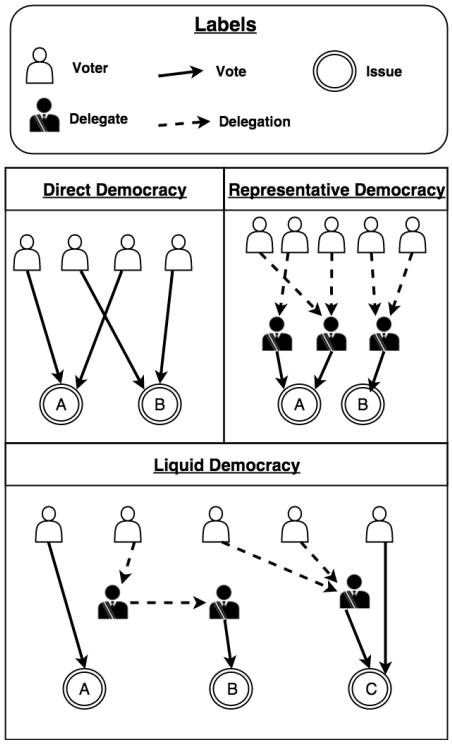

Figure 1: An illustration of the differences between direct, indirect, and liquid democracies.

COVID vaccination status. With vaccination cards, adversaries can track the movements of individuals who received or did not receive specific vaccinations.

Anonymous credentials address this issue by preventing adversaries from tracking any individual's aliases across multiple uses. They also allow communicators to prove aspects of their identity without revealing more sensitive information, often using zero-knowledge proofs as a base.

1.2 Delegateable Anonymous Credentials

However, anonymous credentials alone still don't allow privacy with shareable credentials. Suppose an organization is structured with administrators who can authorize credential-giving power to other organizational members, effectively delegating credential-giving privileges. Using an anonymous certificate and signature system tags the administrators' identities to the members they authorized, indicating a distinct lack of confidentiality.

Despite the additional complexity, researchers have designed a secure and efficient system for delegatable anonymous credentials using Mercurial Signa-

tures. For our project, we will expand upon this research for delegation of democratic representation. That is, we will be designing an anonymous credentials encryption scheme that allows an individual to delegate their vote to another user. Below, we will outline the security policy such a system should satisfy. Our final project will then focus on designing a system to meet the security policies and, if time allows, include an implementation of the designed encryption scheme.

2 Security Policy

To begin the discussion of the system, we will define a number of parties that will participate in the voting process. Namely:

1. **The Central Authority (CA)**: a trusted third party that knows a public key of each Citizen. The CA is able to issue credentials to each citizen. In this case the CA would be the government or other authority facilitating the democratic process.

2. **The Tallying Authority (TA)**: a trusted third party distinct from the CA that will be responsible for receiving credentials from the voting population and tallying the votes.

3. **Citizens**: a subset of the voting population that can only receive votes from the CA directly. A Citizen can choose to either cast their vote directly to the TA, or to vote indirectly by delegating their voting power to a Representative that they trust to act in their best interest.

4. **Representatives**: a subset of the voting population that can only receive delegated votes from Citizens. A Citizen delegates their votes to a trusted Representative, and a Representative will be responsible for voting on behalf of the Citizens they represent

Some key desired properties of our system is as follows:

1. **Minimal Change**: The structure of the system should involve minimal change to the existing democratic structure of the US. For instance, existing representatives would become the "Representatives" of our system.

2. **Anonymity**: The recipient of a delegated vote cannot determine the identity of the source of the vote. That is, a Representative cannot tell which Citizen was the source of a vote.

3. **Knowledge of Representative Power**: The recipient of delegated voting power should be aware of the number of people they represent. This helps this voter know the gravity of their decision.

Additionally, for the sake of completeness, a number of security policies for our system are as follows:

1. The CA can grant the authority to cast a vote to any and all citizens.

2. Once a member of the voting population, Citizen or Representative, casts a vote, that voter cannot convince the TA to record another vote with the same authority and have it count as multiple submissions. In short, a voter should not be able to increase their voting power.

3. A member of the voting population, Citizen or Representative, cannot convince the TA that they have a vote they do not have.

3 Related Work

Before diving into the anonymous delegated credentials literature, it is worth noting that one can design a system similar to the one we are designing

There are many publications relevant to anonymous credentials, with different perspectives, approaches, and building blocks. The construction of anonymous credentials schemes usually rely on a combination of cryptographic primitives, such as zero-knowledge proofs, commitment schemes, and blind signatures.

Anonymous credentials were first conceived by in [Cha83], as a means for an untraceable payments system. David Chaum also introduced the concept of blind signatures, by giving the familiar analog of carbon paper lined envelopes. Blind signatures are similar to the digital signatures, but they additionally allow for messages to be signed without the signer learning the message content. Blind signatures can be designed using a number of public-key encryption protocols, such as RSA encryption.

Baldimtsi and Lysyanskaya [BL13] proposed, constructed, and proved the security of an anonymous credentials scheme using blind signatures, under the DDH assumption. The motivation is to allow users to be able to prove possession of credentials regarding certain attributes (e.g., age), without disclosing any unnecessary information about themselves. To that end, they define blind signatures with attributes, which they use as a building block for their anonymous credentials construction.

The first proposals and constructions of Delegatable Anonymous Credentials (DAC) were often relying on the use of heavy machinery, such as Non-Interactive Zero-Knowledge (NIZK) Proofs and Growth-Sahai commitments. In [CL06], Chase and Lysyanskaya formally define signatures of knowledge, closely related to proofs of knowledge for any NP statement, and they provide relevant constructions to obtain the first delegatable anonymous credentials system.

In [CL20], Crites and Lysyanskaya formally define and construct a Mercurial Signatures scheme for variable-length messages. Building on [FHS14] which constructs structure-preserving signatures on messages of the same equivalence class, [CL20] constructs a scheme allowing a signature σ on a message m under a public key pk to be transformed into a signature σ' for an equivalent message m', under an equivalent public key pk'.

In [CL18], Crites and Lysyanskaya define, construct and prove the security of a delegatable anonymous credentials scheme using mercurial signatures as

their main building block. This involves a root authority (CA) and two main protocols: "Issuing a Credential" and "Proof of Possession of a Credential".

- **Issuing a Credential:** This is an interactive protocol between an Issuer and a Receiver. The Issuer, known under pseudonym nym_I, possesses a credential cred_I at level $L_I(pk_0)$ from the CA. The Receiver is known under nym_R and wishes obtain a credential from the Issuer. This protocol results in the Issuer issuing a delegated credential cred_R to the Receiver, which is at level $L_R(pk_0) = L_I(pk_0) + 1$ from the CA.
- **Proof of Possession of a Credential:** This is an interactive protocol between a Prover and a Verifier. The Prover tries to convince the Verifier that his delegated credential cred_P at level $L_P(pk_0)$ is a valid certification chain under the CA. The result of this protocol is an accept/reject output from the Verifier.

4 Assumptions

In designing our system, we made a number of assumptions to reduce the scope of our design and best illustrate the value of DACs in creating Delegatable Anonymous Voting Systems.

4.1 Authorities are Honest but Curious

The first assumption is that the trusted authorities are honest but curious. In short, this means that the authorities will always perform the stated protocols correctly and honestly. However, we cannot assume that authorities will not synthesize information from the data that the authorities have access to. This assumption motivated the decision of creating two trusted authorities with access to different pieces of information to preserve anonymity of who is voting and whether a particular Citizen voted directly or delegated their vote.

4.2 Hardware and Software are Correct and Secure

Another assumption is that the Hardware and Software is correct and secure. All channels of communication can be implemented using anonymous channels and we will assume that the implementation of DACs are correct. We will also be assuming that their is a secure method of collecting and counting votes, which is not an unreasonable assumption as system designs for correct and verifiable voting systems in literature. This allows us to focus on augmenting such systems with anonymity and the option to vote directly or indirectly.

5 System Description

Our system will include two trusted authorities – the *Central Authority (CA)* and the *Tallying Authority (TA)* – and two categories of voters – *Citizens* and

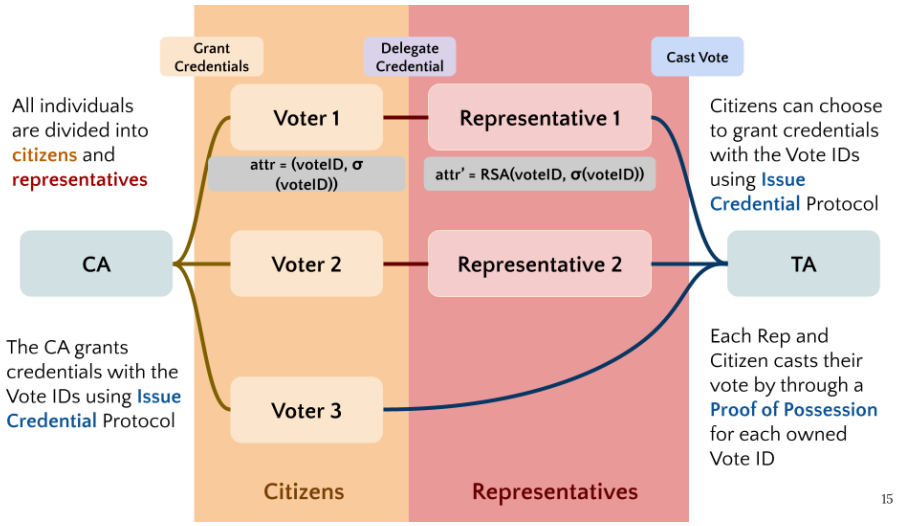

Figure 2: A Diagram of the four parties and the protocols that they can perform.

Representatives. In the following sections, we will outline the responsibilities and protocols that each of these members perform.

5.1 Central Authority (CA) and Tallying Authority (TA)

At the beginning of the voting process, each Citizen is known to the CA by their respective public key. The CA will then generate a unique Vote ID, generated as a UUID with no correlation to the recipients. The CA will then delegate a credential with this Vote ID as an attribute to each Citizen, and only to Citizens.

The CA then sends the list of all the generated, valid Vote IDs to the TA, excluding the public keys of the Citizens that received them. This list of valid Vote IDs is encrypted using a secure asymmetric encryption scheme on the TA's public key to prevent an adversary from reading the list.

At the end of the voting process, the TA will engage in a Proof of Possession with all willing voters that want to cast their ballot(s). This can be Citizens casting their ballot directly, or Representatives casting each ballot that they have received. If there are multiple credentials that carry the same Vote ID, then the TA will discard all credentials of length greater than the minimum delegation length. Then, of the votes with the same Vote ID and same length, the TA will allocate this Vote ID to the option with the plurality. The TA will break the tie by selecting the most recent cast of the options.

5.2 Citizens and Representatives

As mentioned previously, the voting population will be divided into two groups: *Citizens* and *Representatives*.

Each Citizen will receive one credential from the CA to begin with. These Citizens may choose to either delegate their vote to one and only one Representative, or cast their vote directly through engaging in the Proof of Possession protocol with the TA.

In contrast, Representatives do not receive credentials from the CA. A Representative can only receive delegated votes from Citizens (and not other Representatives). Additionally, Representatives cannot delegate their vote to anyone. When it comes time to vote, the Representative will engage in a Proof of Possession protocol with the TA for *each* of the credentials that the Representative has received. Effectively, the Representative's vote is weighted by the number of received votes, correlating with the number of Citizens that trust the opinion of the Representative.

5.3 Delegation Protocol

Whenever a Citizen wants to delegate a credential to a Representative, both parties will engage in the Issue Credential protocol for each of the credentials in the Citizen's possession. A Citizen cannot delegate to multiple Representatives, and once a Citizen has delegated their voting power, their vote cannot be delegate any more.

5.4 Casting Vote Protocol

Whenever a member of the voting population, Citizen or Representative, wishes to cast their vote, they engage in a Proof of Possession protocol with the TA. This protocol will convince the TA that the Citizen or Representative is in possession of a valid credential with the proposed VoteID if and only if the voter actually has a valid credential with the proposed VoteID. This is correct by virtue of the correctness of Proof of Possession in the DAC implementation.

5.5 Revoking Protocol

In addition to being able to delegate, a Citizen can Revoke their delegation only by voting directly. The TA will receive two credential chains with the same VoteID. The TA will then discard the ballot with the VoteID with the longer credential chain and only record the ballot for the VoteID with the shortest VoteID. This allows a Citizen to override a delegated vote anonymously through voting directly.

5.6 Signing and Encrypting VoteIDs

5.6.1 First Attempt and Problems

For our design, we are relying on DACs constructed from Mercurial Signatures for *variable-length messages*, which allows each delegated credential to have an attribute field. In our case, the attribute field must be related to the `VoteID` of the vote being delegated, so that the TA can properly match and count votes.

However, if we simply set `cred.attr = VoteID`, this introduces forgery risk and compromises anonymity.

Forgery: Consider a malicious citizen c who has been issued a credential by the CA with attribute `VoteID`. Since we cannot restrict users from tampering with their credentials, the citizen can just replace `VoteID` with any other `VoteID'`. If that corresponds to the vote ID of another citizen c', then the TA will accept c's credential as valid, allowing c to vote on behalf of c'. Even if we make the space of voteIDs large enough, we can see that a malicious citizen can perform a birthday attack and hit another citizen's voteID with decent probability. This forgery attack can be performed by representatives as well, which can be argued to be even more compromising.

Anonymity and extortion: Suppose we set `cred.attr = VoteID` and leave it unchanged when delegating and/or when casting the vote. Then anyone monitoring the occurring delegations can tie a delegated vote to the citizen who delegated it, since the VoteID is the same. There is also room for extortion; suppose a malicious representative extorts a citizen with vote ID `VoteID` to delegate his vote to them. If the citizen decides to override the extortion by directly casting a vote, the representative might see that vote being cast (also with `VoteID`) and recognize that the citizen overrode them. In short, keeping the plain VoteID field unchanged essentially compromises the anonymity we wanted to guarantee by using DACs in the first place.

5.6.2 Solution

To overcome these issues, we essentially sign and encrypt the VoteID, as also shown in Figure 1. We assume both the CA and the TA have (extra) separate key pairs (sk_{CA}, pk_{CA}) and (sk_{TA}, pk_{TA}), respectively. We also make use of a standard EUF-CMA signature scheme (Gen', Sign, Verify) and a CCA-secure encryption scheme (Gen, Enc, Dec), such as RSA with OAEP. When the CA issues a vote credential to a citizen, it will first produce $\sigma = \text{Sign}(sk_{TA}, \text{VoteID})$ and then set `cred.attr = (VoteID, σ)`. If the citizen wants to directly cast their vote, they do not change the credential attribute. If the citizen wishes to delegate their vote to a representative, they first encrypt `ct` $= \text{Enc}((\text{VoteID}, \sigma))$ and then set the new attribute of the delegated credential `cred'` to be `cred'.attr = (VoteID, ct)`.

When the TA receives a vote, it first checks the level of the credential chain. If the level is $L = 1$ (directly cast vote), then the TA checks that (VoteID, σ) is a valid message-signature pair under public key pk_{CA} using the Verify algorithm. If not, the TA disregards the vote. If the level is $L = 2$ (delegated vote), then the TA first decrypts `ct` to obtain (VoteID, σ) $= \text{Dec}(sk_{TA}, \texttt{ct})$, and then performs the same signature check as in the $L = 1$ case. If the ciphertext is not well-formed and thus the Dec algorithm cannot decrypt, then the TA disregards the vote.

If any citizen tampers with their VoteID, then the signature is not going to be valid anymore, and the TA will reject the vote. Thus security against

forgery largely follows from EUF-CMA security of the signature scheme used. Assuming CCA-security of the encryption scheme used, then no representative can tie the ciphertext they receive with any VoteID being used to directly cast a vote. Further, the representative cannot tamper with the ciphertext to produce an encryption of another VoteID-signature pair, again assuming CCA security. This takes care of the **anonymity** and **extortion** attacks outlined above.

6 Adversarial Model

Our system is robust against a variety of new adversarial attacks that arise in the context of delegatable e-voting, including forgery, double voting, tracking, and extortion.

6.1 Forgery

Suppose an adversary attempts to forge a new vote to increase their voting power by generating a new VoteID. However, the VoteID is encrypted and signed by the Central Authority using the Hash and Sign with the RSA signature scheme. As a result, the adversary's attempt to generate a fresh VoteID only succeeds with negligible probability assuming the Random Oracle Model and assuming that the RSA Problem is hard.

Another attack is plausible in the construction of Mercurial Signatures for various attributes. However, in the context of our voting scheme, this attack is rendered obsolete. The attack is as follows:

Given five voters in the voting system A, B, C, D, and E where A and B both delegate their vote to both C and C delegates their votes to both D and E, D and E can collude to change their given certificates . Suppose that B and C collude to share their certificates. D has a certificate with the signatures $(pk_D, pk_C, pk_A, pk_{CA}), (\sigma_{C\to D}, \sigma_{A\to C}, \sigma_{CA\to A})$ and E has $(pk_E, pk_C, pk_B, pk_{CA}), (\sigma_{C\to E}, \sigma_{B\to C}, \sigma_{CA\to B})$. D can share their credentials with E and E can form a forged credential from A by combining their chain with a prefix of C's chain. The resulting credential chain is: $(pk_E, pk_C, pk_B, pk_{CA})$, $(\sigma_{C\to E}, \sigma_{B\to C}, \sigma_{CA\to B})$, a valid forged credential chain. This would normally be problematic, as D and E can be given credentials with different attributes that are now shared. However, our system directs voters to delegate all their given credentials to each delegatee. If a voter disregards this norm, the system only performs as expected with credential-sharing. Otherwise, the system would be similar to the same system where C withholds votes, which is an issue already prevalent and attributed to individuals in existing voting delegation.

6.2 Tracking

Assume that an adversary attempts to track the origin of a certificate and VoteID pair and unveil the identity of a voter. By the construction of our system, the public keys are obscured at each delegation step by the *changerep* function. Assuming the Public Key Class-Hiding property of Mercurial Signatures [CL20], our certificates are equally untrackable. To prevent tracking the VoteID, the VoteID object is encrypted at each delegation step using the Randomized RSA encryption scheme. Because all CPA signature schemes such as Randomized RSA produce a distribution of ciphertext that are computationally indistinguishable from a another ciphertext distribution, an adversary cannot feasibly detect the path of VoteIDs through the voting network.

6.3 Extortion

We define extortion in e-voting as an adversary threatening a voter into delegating their vote to them, effectively increasing their influence. Recall that, in our system, the Tallying Authority checks all duplicate VoteIDs and only considers the most recent VoteID from the shortest chain. Given this, the victim of extortion casts their own vote directly to the tallying authority and overwrites the vote cast by the adversary, nullifying the attack.

References

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light, 2013.

[Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. pages 78–96, 08 2006.

[CL18] Elizabeth C. Crites and Anna Lysyanskaya. Delegatable anonymous credentials from mercurial signatures. Cryptology ePrint Archive, Report 2018/923, 2018. `https://ia.cr/2018/923`.

[CL20] Elizabeth C. Crites and Anna Lysyanskaya. Mercurial signatures for variable-length messages. Cryptology ePrint Archive, Report 2020/979, 2020. `https://ia.cr/2020/979`.

[FHS14] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. Cryptology ePrint Archive, Report 2014/944, 2014. `https://ia.cr/2014/944`.

[Sch15] Dominik Schiener. Liquid democracy: True democracy for the 21st century. Medium November 23, 2015, 2015. `https://medium.com/organizer-sandbox/liquid-democracy-true-democracy-for-the-21st-century-7c66f5e53b6f`.